

Universidad Mariano Gálvez De Guatemala Sede Portales
Facultad de Ingeniería en Sistemas
Ing. Jonathan Josué García Cuque
Programación II



PROYECTO INDIVIDUAL 2

Eliezer Emanuel Pú Sontay

Carné 9989-23-7954

Cuarto Semestre

Sección "C"

Guatemala, 09 de Octubre del 2024

Sistema de Ingreso de Notas

El proyecto consiste en una aplicación para gestionar las notas de estudiantes. Utiliza Java con Spring Boot para el backend, React para el frontend y MySQL como base de datos. La aplicación permite registrar puntajes de diferentes evaluaciones. Calcular automáticamente el puntaje total y almacenar toda la información en la base de datos y ofrece una interfaz para ingresar y visualizar las notas de manera sencilla.

1) Funcionalidad del Backend

- **Gestión de Estudiantes:** Permite agregar, consultar, actualizar y eliminar los registros de estudiantes y sus notas.
- **Validaciones:** Se asegura de que los puntajes ingresados no superen los límites permitidos para cada evaluación: actividades, parciales y examen final.
- **Cálculo de Puntaje Total:** Calcula automáticamente el puntaje total sumando los puntajes de las diferentes evaluaciones y actividades.
- **Integración con MySQL:** Utiliza MySQL para guardar la información de los estudiantes de manera permanente.

2) Funcionalidad del Frontend

El frontend está desarrollado en React.

- **Formulario de Ingreso de Notas:** Los usuarios pueden ingresar los datos de los estudiantes y sus puntajes en un formulario sencillo.
- **Visualización de Estudiantes:** Después de agregar datos, los usuarios pueden hacer clic en un botón para ver la lista de estudiantes en una tabla clara.

3) Cálculo de Puntaje Total y Validaciones

La aplicación calcula el puntaje total automáticamente al agregar un estudiante, sumando los puntajes de las distintas evaluaciones:

- **Actividades:** Máximo 35 puntos.
- **Primer Parcial:** Máximo 15 puntos.
- **Segundo Parcial:** Máximo 15 puntos.
- **Examen Final:** Máximo 35 puntos.

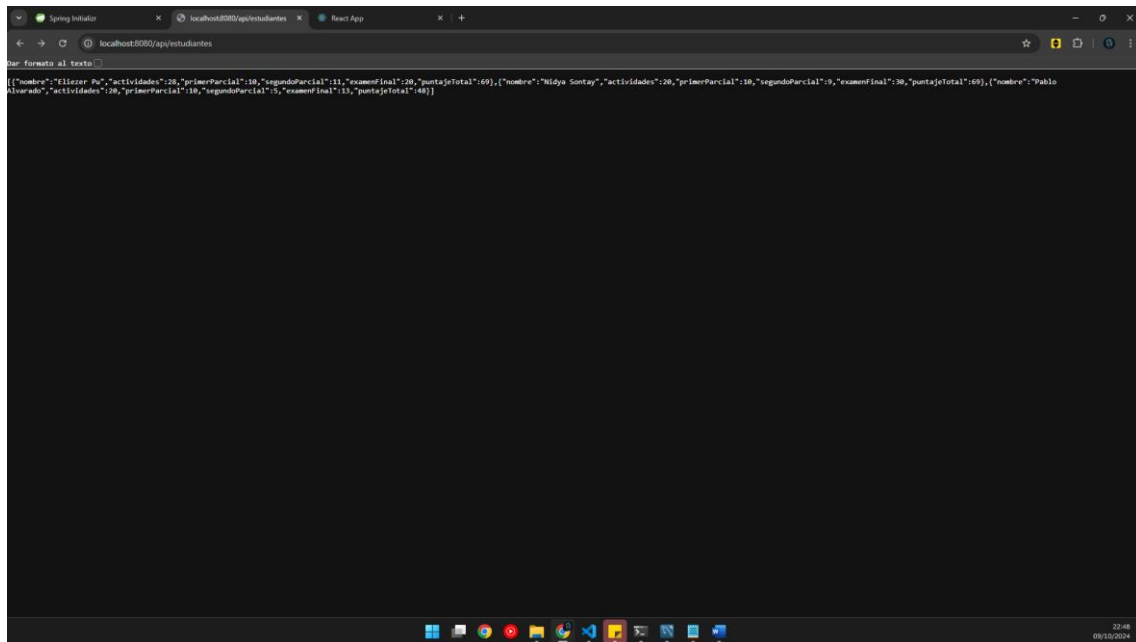
4) Diseño de Base de Datos y Almacenamiento

La base de datos se gestiona en MySQL y tiene una tabla llamada `estudiantes` con los siguientes campos:

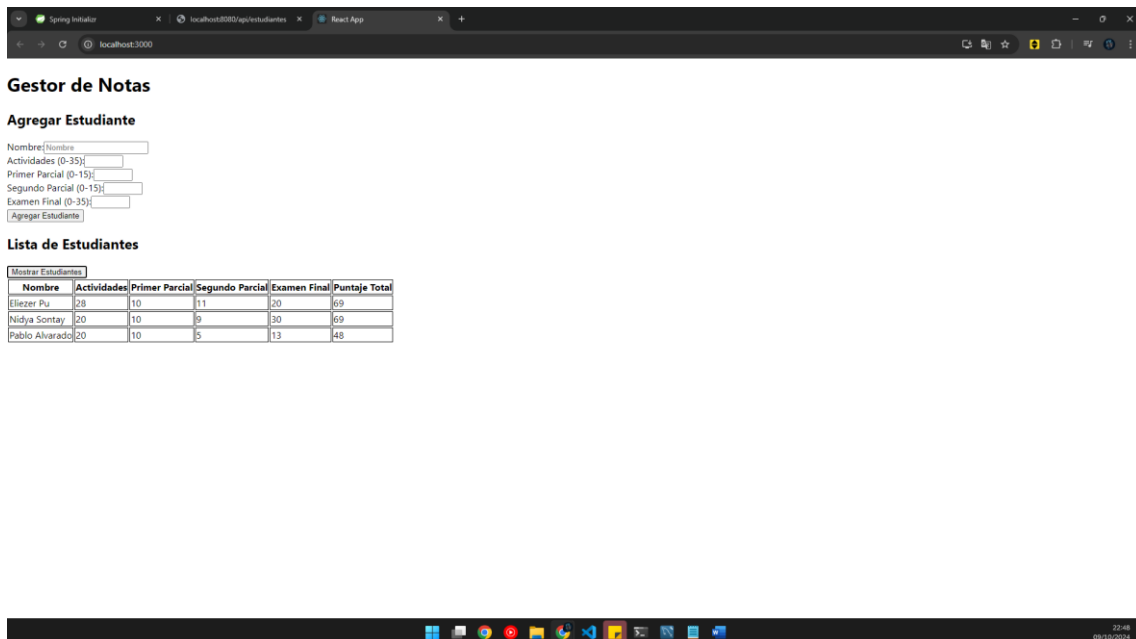
- **id:** Un identificador único para cada estudiante (se incrementa automáticamente).
- **nombre:** El nombre del estudiante (texto).
- **actividades:** Puntaje de actividades (número).
- **primerParcial:** Puntaje del primer parcial (número).
- **segundoParcial:** Puntaje del segundo parcial (número).
- **examenFinal:** Puntaje del examen final (número).
- **puntajeTotal:** Puntaje total calculado (número).

Imágenes del proyecto

Backend



Frontend



Base de Datos

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a tree view of the database structure, including tables like 'estudiante' and 'gestor_notas'. The main editor window shows the SQL script for creating the 'gestor_notas' database and the 'estudiante' table. The table 'estudiante' has columns: 'id' (INT, AUTO_INCREMENT, PRIMARY KEY), 'nombre' (VARCHAR(100), NOT NULL), 'actividades' (INT, CHECK (actividades <= 35)), 'primer_parcial' (INT, CHECK (primer_parcial <= 15)), 'segundo_parcial' (INT, CHECK (segundo_parcial <= 15)), and 'examen_final' (INT, CHECK (examen_final <= 35)). Below the SQL editor, the 'Result Grid' shows the data for the 'estudiante' table. The table has 3 rows of data. The bottom panel shows the 'Action Output' with a list of SQL statements and their execution results.

```
1 CREATE DATABASE `gestor_notas` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;
2 CREATE TABLE `estudiante` (
3   `id` INT AUTO_INCREMENT PRIMARY KEY,
4   `nombre` VARCHAR(100) NOT NULL,
5   `actividades` INT CHECK (actividades <= 35),
6   `primer_parcial` INT CHECK (primer_parcial <= 15),
7   `segundo_parcial` INT CHECK (segundo_parcial <= 15),
8   `examen_final` INT CHECK (examen_final <= 35)
9 );
10 SHOW TABLES;
11 SELECT * FROM estudiante;
12 ALTER TABLE estudiante ADD COLUMN puntajeTotal INT;
13 ALTER TABLE estudiante DROP COLUMN puntajeTotal;
```

| id | actividades | examen_final | nombre | primer_parcial | segundo_parcial | puntaje_total |
|----|-------------|--------------|----------------|----------------|-----------------|---------------|
| 1 | 28 | 20 | Eliezer Pu | 10 | 11 | 69 |
| 2 | 20 | 30 | Nidia Santay | 10 | 9 | 69 |
| 3 | 20 | 13 | Pablo Alvarado | 10 | 5 | 48 |

Columns:

- id: bigint(11)
- actividades: int
- examen_final: int
- nombre: varchar(100)
- primer_parcial: int
- segundo_parcial: int

Action Output:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 34 | 21:52:18 | SELECT * FROM estudiante LIMIT 0, 1000 | 7 row(s) returned | 0.000 sec / 0.000 sec |
| 35 | 21:53:00 | SELECT * FROM estudiante LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 36 | 21:57:30 | TRUNCATE TABLE estudiante | 0 row(s) affected | 0.047 sec |
| 37 | 21:57:33 | SELECT * FROM estudiante LIMIT 0, 1000 | 0 row(s) returned | 0.000 sec / 0.000 sec |
| 38 | 22:16:19 | SELECT * FROM estudiante LIMIT 0, 1000 | 0 row(s) returned | 0.000 sec / 0.000 sec |
| 39 | 22:49:27 | SELECT * FROM estudiante LIMIT 0, 1000 | 3 row(s) returned | 0.000 sec / 0.000 sec |

Link de git: <https://github.com/Eliezer130502/Sistema-de-Ingreso-de-Notas.git>

En archivo estará un bloc de notas con nombre Readme. Esto servirá como indicaciones para probar el programa.